Received: 5 April 2016 | Revised: 18 October 2016 | Accepted: 7 March 2017

DOI: 10.1111/coin.12128

ORIGINAL ARTICLE

WILEY Computational Intelligence

Learning over subconcepts: Strategies for 1-class classification

Shiven Sharma[1] | Anil Somayaji[2] | Nathalie Japkowicz[1]

[1]School of Information Technology and Engineering, University of Ottawa, Canada

[2]School of Computer Science, Carleton University, Canada

Correspondence
Shiven Sharma, School of Information Technology and Engineering, University of Ottawa, Ontario, Canada, K1N 6N5.
Email: shiven.cheema@gmail.com

Present Address
Nathalie Japkowicz, American University, 4400 Massachusetts Avenue, NW Washington, DC 20016

Funding information
Canadian government through an NSERC/ISTP CRD

Abstract

In machine learning research and application, multiclass classification algorithms reign supreme. Their fundamental property is the reliance on the availability of data from all known categories to induce effective classifiers. Unfortunately, data from so-called real-world domains sometimes do not satisfy this property, and researchers use methods such as sampling to make the data more conducive for classification. However, there are scenarios in which even such explicit methods to rectify distributions fail. In such cases, 1-class classification algorithms become the practical alternative. Unfortunately, domain complexity severely impacts their ability to produce effective classifiers. The work in this article addresses this issue and develops a strategy that allows for 1-class classification over complex domains. In particular, we introduce the notion of learning along the lines of underlying domain concepts; an important source of complexity in domains is the presence of subconcepts, and by learning over them explicitly rather than on the entire domain as a whole, we can produce powerful 1-class classification systems. The level of knowledge regarding these subconcepts will naturally vary by domain, and thus, we develop 3 distinct methodologies that take the amount of domain knowledge available into account. We demonstrate these over 3 real-world domains.

KEYWORDS

anomaly detection, classification, machine learning, 1-class classification

1 | INTRODUCTION

With the proliferation of data collection over the past few decades, the ubiquity of data in most domains provides an exceptionally conducive platform for practical research into the application of machine learning algorithms, particularly classification. In particular, interest from industry is increasing in using intelligent algorithms for fast and efficient solutions to tasks that would typically require vast human resources or cumbersome and outdated approaches. A tremendous amount of data is now being collected at a very fast rate, and there is a strong desire to *make sense* of these massive data sets by using techniques from machine learning.

However, with this exposure of machine learning to these data sets, many of the standard, implicit assumptions that machine learning methods often make are now being torn apart. This has a direct consequence on how algorithms and ideas from machine learning can be applied to these domains. Specifically, algorithms that not only handle the issues associated with massive data sets but are also able to take advantage of the nuances inherent in them to ameliorate performance need to be tuned and developed.

There are 2 significant facets to massive data sets in particular that have the most severe impact on machine learning solutions. The first, and perhaps most significant and pervasive issue, is that of *class imbalance*. Most of the binary and multiclass classification algorithms that form the core of the machine learning world assume relatively balanced distributions of categories (classes) to induce appropriate discriminant functions. Put more simply, there are sufficient data from all known categories that make up the domain to build a model that can classify novel data with acceptable accuracy. But, more often than not, real-world data are far from balanced; there is, typically, an overabundance of data from certain classes, and there are very little data from other classes. The issues arising directly, or indirectly, from such imbalanced distributions deserve special attention and, of late, research into learning in such domains has gained in popularity[1]; oversampling, undersampling, cost-sensitive classification, etc are some of the conventionally used methods used for rectifying imbalance. However, there are situations, for example, in security domains and text classification, when imbalances are of such an extreme nature that even methodologies designed explicitly for handling them cannot be applied. In particular, the only data available for learning a classification system are from a single class (or a group of similar classes); data from other classes are either nonexistent or impossible to collect or exceptionally rare. In such scenarios, 1-class classification, a special paradigm of classification algorithms, becomes the only viable framework for learning.

In our past work, we explored in detail the problem of which paradigm of learning, 1 class or binary/multiclass, to use, based on the levels of imbalance.[2] Establishing the situations under which 1-class classification is the suitable choice is one matter; actually using it is an entirely different matter altogether. This leads us to the second issue, one that directly impacts 1-class classification, namely, domain complexity. Massive data sets tend to have a highly complex distribution; the entire data space can, in effect, be clustered into distinct subconcepts, each of which forms their own unique space. Modeling a single class that exhibits a complex distribution is a typically hard task; the advantage that 1-class classifiers possess over multiclass classifiers in domains of imbalance disappears when the domain in consideration is complex. We hypothesize that the complexities arise because of the presence of subconcepts, and depending on the domain and availability of domain experts, we can have varying amounts of knowledge regarding what the subconcepts are. In the ideal case, we have complete knowledge regarding what the subconcepts are and are able to identify them during classification. However, in other domains, we may need to learn them in a supervised manner with the aid of a domain expert. The worst case occurs when we have no knowledge and must discover them via unsupervised learning. Regardless of the amount of knowledge, the core idea is to simplify the original domain into subdomains that are easier to model; learning in the context of subconcepts rather than over the entire domain will narrow down the focus of the learner, allowing it to be more effective at accepting target class instances and rejecting data from all other classes.

Our work was inspired by research conducted over 3 real-world domains. The first domain comprised data representing Xenon isotopes and dealt with the compliance verification of the Comprehensive Test Ban Treaty (CTBT)[3,4]; we were tasked with investigating whether machine learning could be applied so as to automate the detection of clandestine nuclear tests by nations. The second domain comprised gamma ray spectra and involved investigating the applicability of machine learning for the purposes of detecting gamma ray signatures emitted from dangerous isotopes, for example, uranium or plutonium. The third domain related to biometric sensor data gathered from a mobile phone's accelerometer, gyroscope, and touch screen during a swipe action, the aim being to use this to authenticate only the user of the phone. All these domains suffer from extreme imbalance, and this precludes the application of binary or multiclass classification algorithms.

The article is structured as follows. Section 2 will provide an overview on the various methods used for learning over complex domains exhibiting extreme class imbalance. Section 3 introduces the strategies for improving 1-class classifier performance by dividing the domain along subconcepts. An empirical analysis is conducted over artificial and UCI data sets, as well as the 3 real-world domains that motivated our research. We begin by validating our strategies over the artificial and UCI data sets in Section 4. This is followed by an application of our strategies over the 3 domains in Section 5. Section 6 provides concluding remarks, and possible directions for future research are detailed in Section 7.

2 | CHALLENGES IN IMBALANCED DOMAINS

In this section, we review research done within the field of extreme imbalance. Extreme imbalance warrants the application of 1-class classifiers, as explored in the work of Bellinger et al,[2] and thus, our review will focus only on research done under 1-class classification. For a comprehensive review on research done in the general area of imbalance, we direct the interested reader to the excellent survey by He and Garcia.[1]

The 2 aspects that affect the performance of 1-class classifiers considered in this work are overlap and multimodality, both of which contribute to the overall complexity of a domain. It is well established that the degree of overlap between classes severely impacts classifier performance; the more the overlap between classes, the harder the learning task. This holds true for both binary and 1-class classifiers. Furthermore, if a domain is highly multimodal, there is a risk of a 1-class classifier overgeneralizing as it attempts to “cover” all areas of the domain. We begin this section by reviewing work done in understanding the relationship between the level of imbalance and the level of overlap between various classes in domains. This is followed by a review on research done in improving performance over imbalanced, multimodal domains.

2.1 | Imbalance and overlap

Studies in imbalance and overlap have been primarily conducted under a binary or multiclass classification setting. Though no studies exist that explicitly look at the impact of overlap on 1-class classifiers, research done over binary classifiers offers insight into the effects of overlap on 1-class classifiers. Thus, we review the relevant research in this subsection.

While much of the focus in literature has been on rectifying the imbalance explicitly, a few researchers have examined the extent to which imbalance really affects classifier performance. Are there properties of the domain that might make the impact of imbalance less severe? Japkowicz[5] discovered that if the classes are linearly separable, a classifier such as a support vector machine (SVM) would not be impacted by imbalance. Specifically, they note that the problems due to class imbalances are relative to various factors, such as the complexity of the concept (ie, how complex the probability distribution function that generates the data is) and the size of the training set; the classifier performance will be immune to imbalance if the concept is simple and the training set is large. Prati et al[6] conduct a similar study; using 10 artificially generated data sets, each with varying levels of overlap, they observe that as the overlap decreases, the effect of imbalance becomes less severe. To further understand the nuances of overlap and imbalance, García et al[7] examine the performance of classifiers by looking at the overall imbalance ratio in the data (global imbalance), the imbalance ratio in the overlap region (local imbalance), and the amount of data in the overlap region. They observe that a class that is better represented in the overlap regions is better classified by global learners, while a class that is not well represented is classified better by local learners. The dependency between overlap and imbalance is further studied by Denil and Trappenberg,[8] where they observe that given enough data, imbalance has minimal impact, whereas if the overlap is severe, even optimal classifiers will suffer in performance.

All these studies conclude that imbalance is not always the issue; the degree of overlap between the classes has a major impact. In other words, the greater the overlap, the harder the learning task, irrespective of imbalance. Therefore, the impact of overlap between classes traverses both binary/multiclass and 1-class classification paradigms; in the latter, the challenge will be in inducing an appropriate function that does not overgeneralize into the overlap region. Thus, these studies serve to highlight that even in a 1-class classification setting, handling overlap is important in producing accurate classifiers.

In the following subsection, we consider the other facet of data complexity, that of multimodality.

2.2 | Imbalance and multimodality

Multimodality in a domain can arise by the presence of multiple subconcepts. In this section, we review work conducted to improve the performance of 1-class classifiers when faced with multimodal domains. We begin by considering ensemble-based approaches that resample the domain and use existing ensemble learning algorithms, followed by work done under a clustering-based framework that divides the domain space by some form of clustering. We end this subsection by considering research done that exploits known domain knowledge to divide the domain for improving classification.

2.2.1 | Ensemble learning–based approaches

Ensemble methods such as bagging and boosting have had tremendous success in multiclass classification, and thus, it should come as no surprise that methods inspired from these algorithms have found their way within the realm of learning with imbalance. Shieh et al[9] use bagging to create an ensemble of 1-class SVMs (OCSVMs). They note that in the presence of noisy and borderline samples, OCSVM learns an enlarged boundary, which leads to a large number of false positives. They evaluate both a regular OCSVM and the bagged version over artificial and 3 UCI data sets and show that their bagged version outperforms the regular OCSVM in all cases, especially when they introduce noise into the data sets. Desir et al[10] create bootstrap replicas of the training data and generate outliers for each set, thus converting each subset of the data into a binary classification problem. Over these subsets, they train random forest classifiers and aggregate their decision. Through experiments conducted over data sets from the UCI repository, they compare the performance of their method against OCSVM, Gaussian estimator, Parzen windowing, and Gaussian mixture and demonstrate that on most data sets, their approach performs equally well or better than these algorithms.

2.2.2 | Clustering-based approaches

For clustering-based approaches, the idea is to partition the data space into distinct subspaces over which individual 1-class classifiers can be trained. Wang et al[11] exploit the inherent target data structures obtained via hierarchical clustering to create an ensemble of spherical 1-class classifiers. Further work in using clustering for creating 1-class classifier ensembles is conducted by Lipka et al,[12] where the authors use the *k*-means algorithm to create an ensemble of OCSVM classifiers. A general-purpose framework for using clustering for improving 1-class classification is proposed by Krawczyk et al.[13] Their framework consists of 3 parts: the choice of clustering algorithm, the choice of 1-class classifier, and the choice of how to combine decisions. In all studies, the ensemble appears to outperform the single 1-class classifiers. Within the area of network intrusion detection, Leung et al in Leung and Leckie[14] develop a clustering algorithm based on pMAFIA to perform anomaly detection on network data, with the aim to cover 95% of the training data (using the KDD Cup data set), thus making the 5% of data points not covered as outliers.

2.2.3 | Learning with domain knowledge

While the previous sections used generic methods for partitioning the data space, some research, especially in security, has looked into using the nuances of the domain itself for partitioning the space. In other words, rather than simply cluster or sample based on a general heuristic, they look at whether groups can be formed such that all the data in a particular group conform to a predetermined heuristic as determined by a domain expert.

In the domain of handwritten character recognition, a nonclustering-based approach to autoassociation is discussed by Schwenk and Milgram.[15] The authors describe *Diabolo networks*, in which an autoassociator (AA) is trained on a particular class, inherently reducing a multiclass problem into multiple 1-class problems. Each character is treated as a separate class, and a network is trained on it. The class whose network returns the smallest reconstruction error is assigned to the corresponding test instance. Note that the core idea in this approach may not have anything to do with alleviating the problem of small disjuncts but of reducing a complex domain, ie, the set of all characters in the English language as applicable to handwritten character recognition, into simpler domains, ie, the individual characters.

In the domain of network security, Giacinto et al[16] propose a modular system. Specifically, they observe that traffic over a TCP/IP network is made up of packets pertaining to different services, each characterized by its own unique pattern. Thus, it follows that a unique classifier must be induced for each service, rather than the traffic as a whole. They identify 6 services: Web, mail, Internet Control Message Protocol, FTP, Internet Control Message Protocol, and miscellaneous. Furthermore, for each service, they use 3 sets of features: 1 for content specific information, 1 for intrinsic information, and 1 related to traffic. Thus, each service has 3 classifiers trained for it. Using a variety of multiclass classifiers over the DARPA data set, the authors report an improvement in classification performance and a reduction in false alarms. The use of multiclass classifiers is not very practical (as attack data are typically unavailable for learning), and thus, in a follow-up work by Giacinto et al,[17] the authors extend this framework to train 1-class classifiers, specifically, the 1-class SVM, Parzen density estimator, and *k*-means. As with multiclass classifiers, they observe that the ensemble provides superior performance as opposed to just single classifiers.

Neither of the studies discussed in this subsection use a general heuristic. Instead, they divide and conquer based on explicit domain knowledge. The principles, however, are the same: divide the domain space and learn over each subset.

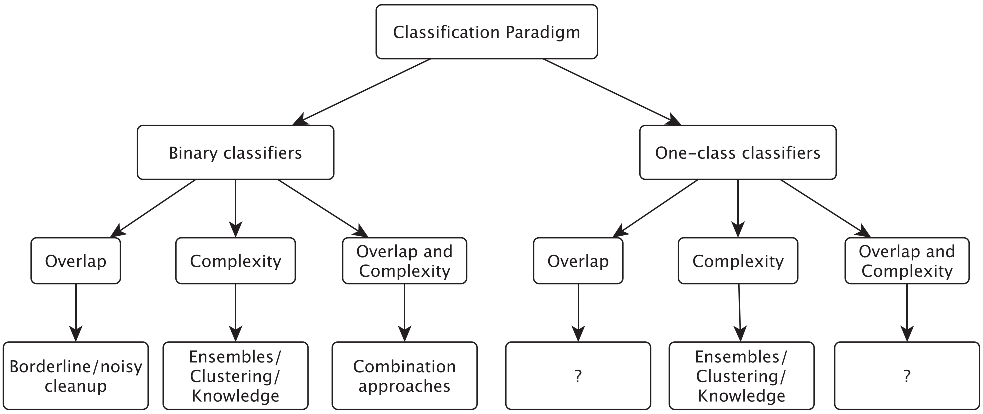


FIGURE 1 An overview of research coverage in the field of learning over imbalanced domains

2.2.4 | Summary

In Figure 1, we provide a graphical overview regarding the various aspects of research in learning over imbalanced domains. We observe that, irrespective of the choice of learning paradigm (binary or 1 class), the challenges in building efficient classifiers are similar: One has to deal with either overlap, domain complexity (eg, due to multimodality), or a combination of both.

As the figure demonstrates, and indeed, from the review presented in this chapter, if we are to use binary classifiers, then the task of rectifying the challenges posed by various aspects of imbalance is relatively easy. It is when the choice is to use a 1-class classifier that we observe a dearth in research and solutions for handling all challenges presented by imbalance. The notion of overlap only really makes sense when we have knowledge of both classes; given only data from a single class, it is difficult to ascertain whether other classes will overlap with the known class. This issue will be further exacerbated if the domain exhibits multimodality. It is this void in research that is addressed in this article.

3 | LEARNING THE SUBCONCEPTUAL LAYER

The performance of 1-class classifiers is severely impacted if the domain under consideration is *complex*. These complexities can arise because of overlap as well as multimodality, and generalizing over them can lead to poor classifier performance. Therefore, if we wish to improve performance, it is imperative to handle these complexities appropriately. A step towards this end is to understand what causes a domain to exhibit them. We propose that a root cause for these complexities is the presence of multiple subconcepts that underlie the domain space, an idea explored by Sharma et al[18]; each of these subconcepts represents a unique property within the overall domain and can typically be identified by an expert in the domain. Thus, we hypothesize that by identifying and isolating these concepts and learning over them individually, one can mitigate the effects of domain complexity and induce more accurate 1-class classifiers. In this section, we formally introduce our strategies for 1-class classification over subconcepts.

We begin by delving into the notions of *main concepts* and *subconcepts*. As an example to illustrate the general idea, let us consider a simple learning problem that involves distinguishing between spoiled beans and fresh beans within the family of *common beans* (*Phaseolus vulgaris*). The domain of common beans has a number of different *aspects*, each corresponding to a type of bean. For simplicity, in our example, we only consider 3 aspects: kidney beans, pinto beans, and white beans.

Our learning task is to learn the *concept* of *fresh beans*. Because of the presence of 3 different aspects, this concept is represented by 3 *subconcepts*: *fresh pinto beans*, *fresh kidney beans*, and *fresh white beans*.

Tfig:conceptTypes illustrates the generalization of our bean example to concepts and subconcepts. In particular, it illustrates a domain with 3 aspects and 2 classes:

- The *target* class is the class over which we will induce a 1-class classifier. In our bean example, this would represent fresh beans. The *outliers* thus would be spoilt beans.
- The different *aspects* of the domain are denoted by A_1, A_2, and A_3. In our bean example, these would be kidney, pinto, and white beans.
- The *concept* that we wish to learn is represented by the target class within each aspect. Thus, the concept is represented by (*Target 1*∪*Target 2*∪*Target 3*), and *Target 1*, *Target 2*, and *Target 3* correspond to the *subconcepts* of the concept. For our bean example, *Target 1* would represent fresh kidney beans, *Target 2* fresh pinto beans, and *Target 3* fresh white beans, the concept to learn being that of fresh beans.

The gradient in the image represents density; darker regions contain the bulk of the data. The image further serves to illustrate how the presence of multiple aspects can cause the domain to exhibit both *multimodality* and *overlap* between the classes.

In the presence of subconcepts, there are 2 approaches that can be used for inducing a classifier to learn the main concept:

- Learn the concept as a whole: In this approach, learning is done for the concept over all aspects of the domain. In our bean example, the training data would consist of fresh beans of all types (aspects).
- Learn over subconcepts: For this approach, learning is done over each subconcept corresponding to the different aspects of the domain. In our bean example, the training data would be divided based on the type of fresh beans and 3 classifiers would be built, 1 for each subconcept.

Thus, in the first approach, we ignore the presence of subconcepts and simply learn the entire concept, whereas in the second approach, we tailor our learning methodology to explicitly handle the different subconcepts. The question thus becomes, which approach to take? Specifically, how does making a conscious decision to learn along subconcepts impact learning?

To answer this question, let us once again consider the domain illustrated in Figure 2. If we are to induce a 1-class classifier over the target class without taking each subconcept into account, we get a classifier as shown in Figure 3. Given the complexity of the domain, the classifier would overgeneralize over the subconcepts, the consequence being that while it would indeed cover most of the target class, it would erroneously classify most of the outlier data as belonging to the target class as well.

If we are to acknowledge the presence of aspects and divide the target data based on the associated subconcepts, on the other hand, we would get a classifier (or a set of classifiers) as shown in Figure 4. By learning over the subconcepts, we do not risk overgeneralization; each classifier focuses only on the targets belonging to a single subconcept. Thus, the resulting classifier would have a much lower error rate over the outliers.

To summarize, we hypothesize that, in the presence of aspects, learning directly over the concept can yield poor performance, especially due to overgeneralization. Using domain knowledge to identify the aspects, and consequently the subconcepts, we can prevent a classifier from overgeneralizing, thus resulting in better classification. In subsequent sections, we will empirically validate this observation. The question now is how to actually identify the aspects. Depending on the nature of the aspects and the associated domain knowledge, we propose 3 different strategies. These are elaborated in the following sections.

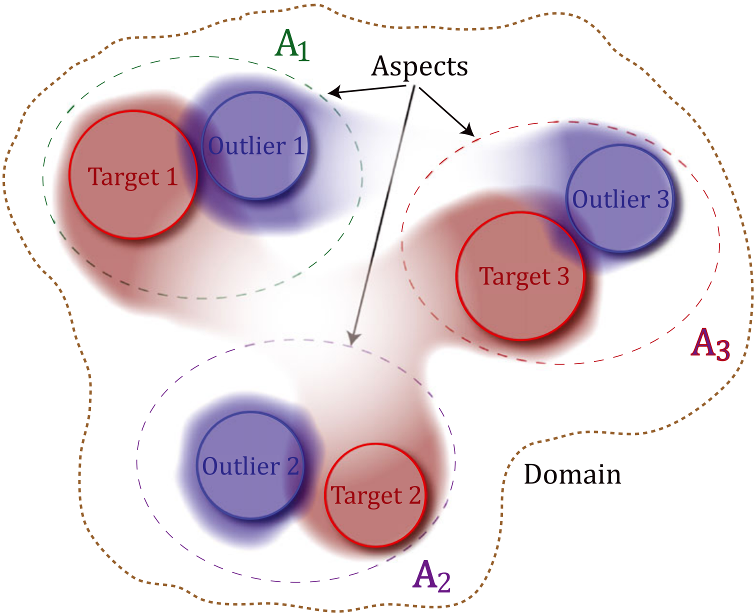


FIGURE 2 The notion of main and subconcepts [Color figure can be viewed at wileyonlinelibrary.com]

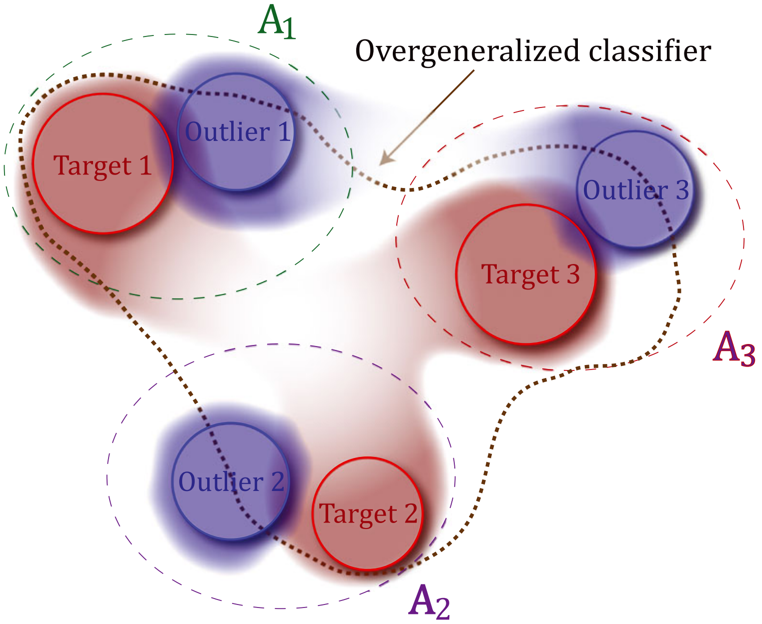


FIGURE 3 Inducing a classifier without distinguishing between different subconcepts [Color figure can be viewed at wileyonlinelibrary.com]

3.1 | One-class classification with complete knowledge

The most ideal case occurs when it is possible to identify, with full confidence and accuracy, which aspect a novel instance from the data space belongs to. With respect to training, a domain

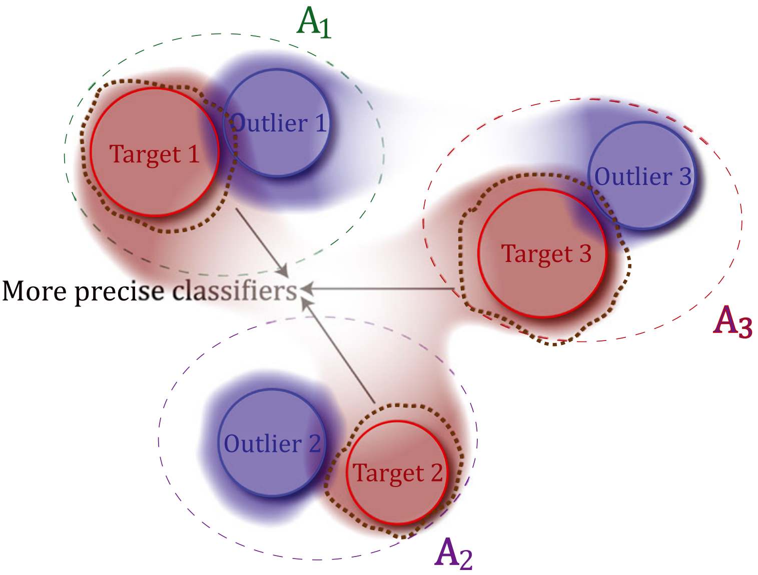


FIGURE 4 Inducing classifiers by distinguishing between different subconcepts [Color figure can be viewed at wileyonlinelibrary.com]

expert is available to identify the different aspects and categorize the training data into appropriate subconcepts, and we can build 1-class classifiers over these subconcepts. Furthermore, once we have built an ensemble of 1-class classifiers, the system knows to which aspect a novel instance belongs. Thus, novel instances can be processed by the 1-class classifier tailor-made for that particular aspect.

This idea is illustrated in Figure 5. Consider a domain in which there are 2 aspects, A_1 and A_2. In practice, the domain expert will provide us with training data from both aspects, from which we induce two 1-class classifiers OC 1 and OC 2 using target data T_1 corresponding to A_1 and T_2 corresponding to A_2. Now, with respect to classification, for every novel instance *i*, we are able to identify which aspect it belongs to. If *i* belongs to A_1, we classify it using classifier OC 1, and by OC 2 if it belongs to A_2.

To summarize, one can use this strategy in the following conditions:

- If we have knowledge of which aspect the training samples belong to
- If we are able to identify which aspect novel testing samples belong to

To illustrate the applicability of the strategy, let us consider our bean example from earlier. As the *aspects* (ie, the type of beans) are fully identifiable, for training, we would know which *subconcept* the training data belongs to, and so we would have 3 training sets for each target subconcept, 1 each for fresh kidney, pinto, and white beans. Thus, we would end up with three 1-class classifiers, each representing a single type of fresh bean. During classification, we would have a mechanism that would be able to tell whether the bean to be classified is a kidney bean, pinto bean, or white bean. The key point to note here is that the mechanism would simply detect the type of bean, and not whether it is fresh or spoilt; that is the task of the 1-class classifiers. In other words, we are identifying the underlying aspects of the domain. Based on the type, the bean would be sent to the appropriate 1-class classifier that would then decide whether it is fresh or spoilt.

This strategy is conducive for the domain of swipes for biometrics considered in our practical case study, detailed in Section 5.1. The underlying aspects pertain to motion and are fully identifiable via the smartphone sensors during both training and classification.

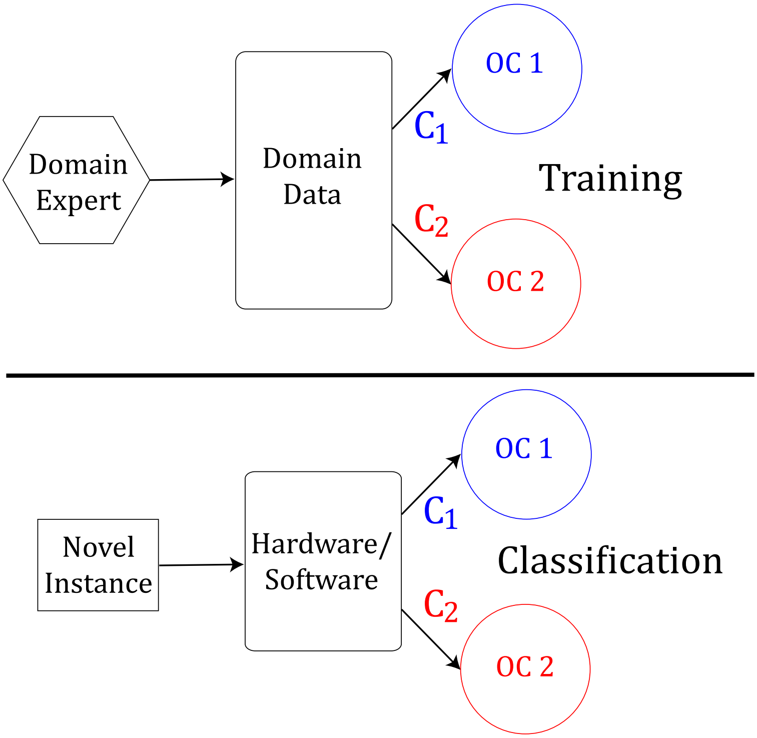


FIGURE 5 One-class classification with full knowledge [Color figure can be viewed at wileyonlinelibrary.com]

3.2 | One-class classification with fuzzy knowledge

In reality, it may or may not be possible to identify the underlying aspects in the domain. This may occur either because of the inability of the available hardware or software to perform such a task without human intervention or because the underlying processes that conform to the aspects themselves are very difficult to quantify; the concepts are *fuzzy*.

For the case of fuzzy knowledge, we propose the following strategy: Training the 1-class classifiers is done as in the strategy for full knowledge by using the target data. However, because the hardware cannot explicitly identify the concept, we induce a multiclass classifier over the known aspects. Depending on how the aspects are represented in the domain, the data used for this may or may not include outlier instances; this is because we are learning to differentiate between the aspects, and not targets and outliers. Now, when novel instances are encountered, they are classified into the appropriate aspect by the multiclass classifier and processed by the appropriate 1-class classifier. The strategy is outlined in Figure 6.

To summarize, one can use this strategy in the following conditions:

- If we have knowledge of which aspect the training samples belong to
- If we are unable to identify which aspect novel testing samples belong to

Let us now consider how this strategy would be applied to our bean example. The *aspects* (ie, the type of beans) in this case would be identifiable during training, but during classification, we would have no way of knowing whether the bean to be classified for freshness is a kidney, pinto, or white bean. Thus, we would first use supervised learning for learning the underlying aspects of the domain to

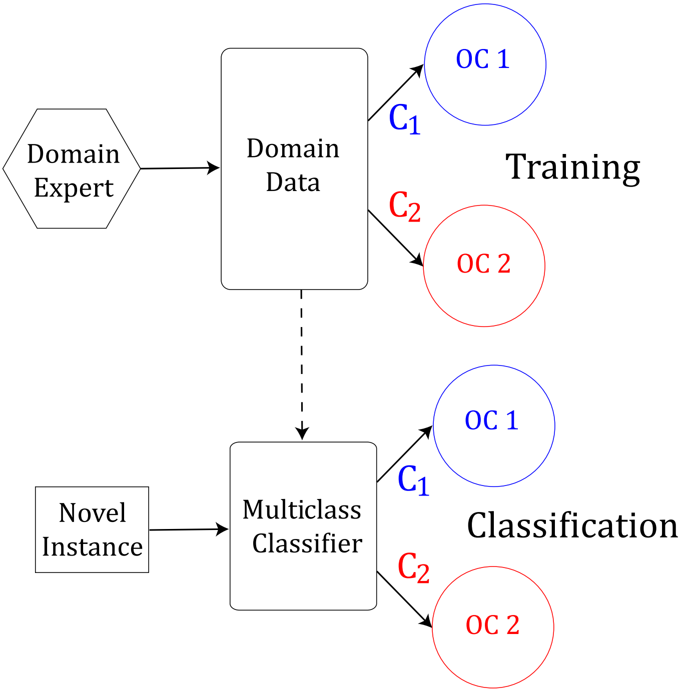


FIGURE 6 One-class classification with fuzzy knowledge [Color figure can be viewed at wileyonlinelibrary.com]

aid us during the classification phase. Specifically, we would have a training set composed of kidney, pinto, and white beans, irrespective of whether they are fresh or spoilt, and we would train a multiclass classifier over the type of beans. For training the 1-class classifiers over the target concept, we would have 3 training sets, 1 for each *subconcept*, corresponding to fresh kidney, pinto, and white beans. Thus, the multiclass classifier learns the aspects of the domain, and the 1-class classifiers learn the target subconcept corresponding to each aspect. During classification, a bean would be first passed to the multiclass classifier to identify its type, and then it would be passed to the appropriate 1-class classifier to ascertain whether it is fresh or spoilt.

This strategy is conducive for the domain of detecting anomalous gamma ray spectra considered in our practical case study, detailed in Section 5.2. The underlying aspects pertain to the presence of water in the environment and are fully identifiable by the physicists at Health Canada during training (as they are able to label spectra accordingly) but cannot be determined during classification.

3.3 | One-class classification with no knowledge

The worst-case scenario occurs when we have no knowledge of the aspects underlying the domain, for example, because of the lack of a domain expert in the field. To divide the domain into the unknown aspects, we turn to classical unsupervised learning methods: clustering. The purpose of clustering is to divide the data space into a number of regions such that instances in a particular region are most similar to each other; this is illustrated in Figure 7. Naturally, instances that are affected by the same aspect will be most similar to each other, and thus, in the absence of knowledge of aspect, we simply cluster the data space and build 1-class classifiers over each cluster, the aim being that each cluster will represent an unknown aspect. The final classifier is an ensemble of all the various classifiers built on the clusters. Classification is done as follows: If an instance is positively classified by at least 1 of the models, then it is assigned to the *target* class; otherwise, it is classified as an *outlier*.

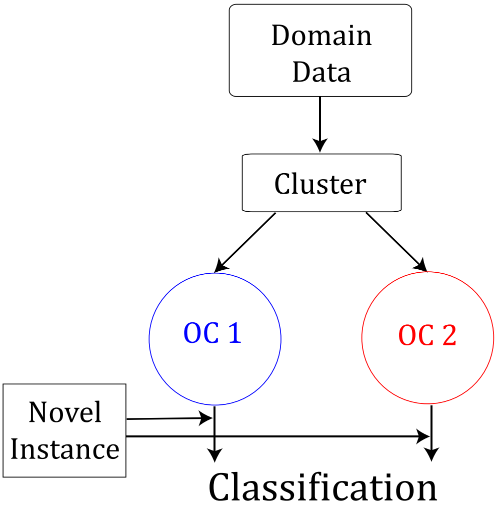


FIGURE 7 One-class classification with no knowledge [Color figure can be viewed at wileyonlinelibrary.com]

Thus, this strategy is to be applied in the following condition:

- When we have no knowledge regarding what aspects occur in the domain

This strategy would be applied to our bean example if we have absolutely no idea as to what type of beans will be given to us. In other words, we have no knowledge regarding the *aspects* of the domain. We would simply cluster the training data of fresh beans and induce 1-class classifiers over each cluster. During classification, each bean would be passed to these classifiers and would be classified as fresh if one of the classifiers deems it to be so.

This strategy is ideal for the domain of classifying a Xenon signature for compliance verification of the CTBT considered in our practical case study, detailed in Section 5.3. While we know the domain is multimodal based on statistical analysis, we have no knowledge regarding any aspects in the domain.

In subsequent sections, we empirically validate the utility of these strategies over complex domains. We first validate them over artificial and UCI data sets. This is followed by their application over the 3 domains encountered during the course of our research work.

4 | VALIDATING THE STRATEGIES

The previous section painted a theoretical picture of our strategies for 1-class classification over complex domains. In this section, we empirically validate them over artificial and UCI data sets. We begin by describing the artificial data sets that we create, followed by the UCI data sets, followed by an overview of the experimental framework. The results are presented at the end of the section.

4.1 | Data description

4.1.1 | Artificial data

The purpose of using artificial data is to create idealized data distributions over which we can control the 2 aspects of complexity considered in this article, namely, multimodality and complexity. In particular, we use three 5-dimensional artificial data sets that are various combinations of multimodal and unimodal target and outlier distributions. We create 2 multimodal distributions, one in which there is no overlap between the modes and the other in which we force overlap. The specifications for these are as follows:

Data 1: Unimodal target and multimodal outlier distributions:

Target: $N([15, 15, 15, 15, 15], 2.75)$
Outlier: $N([5, 5, 5, 15, 15], 2) \cup N([25, 25, 25, 15, 15], 2)$
$\cup N([15, 15, 15, 5, 5], 2) \cup N([15, 15, 15, 25, 25], 2)$

Data 2: Multimodal target and multimodal outlier distributions, no overlap:

Target: $N([5, 5, 5, 5, 5], 3) \cup N([25, 25, 25, 5, 5], 3)$
$\cup N([5, 5, 5, 25, 25], 3) \cup N([25, 25, 25, 25, 25], 3)$
Outlier: $N([15, 15, 15, 2.5, 2.5], 2) \cup N([27.5, 27.5, 27.5, 15, 15], 2)$
$\cup N([2.5, 2.5, 2.5, 15, 15], 2) \cup N([15, 15, 15, 15, 27.5, 27.5], 2)$

Data 3: Multimodal target and multimodal outlier distributions, overlap:

Target: $N([10, 5, 5, 10, 5], 3) \cup N([20, 25, 25, 10, 5], 3)$
$\cup N([10, 5, 5, 20, 25], 3) \cup N([20, 25, 25, 20, 25], 3)$
Outlier: $N([17.5, 15, 15, 5, 2.5], 2) \cup N([25, 27.5, 27.5, 17.5, 15], 2)$

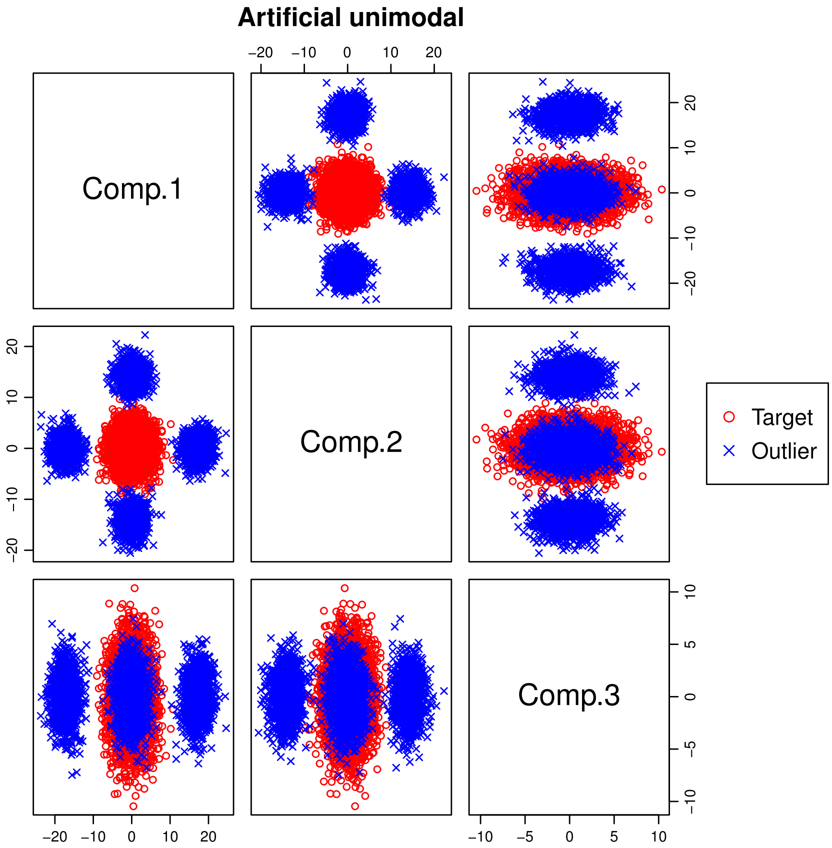


FIGURE 8 The first 3 principal components of the *unimodal artificial* data set [Color figure can be viewed at wileyonlinelibrary.com]

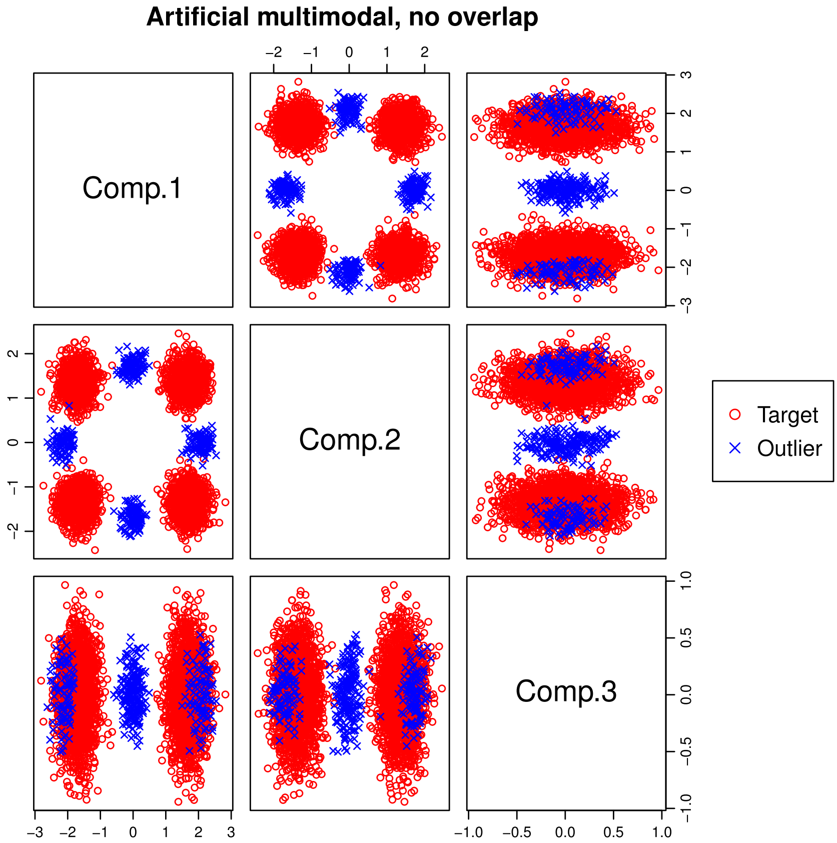


FIGURE 9 The first 3 principal components of the *multimodal artificial* data set with no overlap [Color figure can be viewed at wileyonlinelibrary.com]

$$\cup N([5, 2.5, 2.5, 12.5, 15], 2) \cup N([12.5, 15, 15, 15, 25, 27.5], 2)$$

Figure 8 shows the principal component analysis plots of the first 3 components of the unimodal artificial data set, whereas Figures 9 and 10 display the multimodal artificial distributions without and with overlap, respectively. In all data sets, there are 4000 target and 2000 outlier instances (equally split between each mode).

4.1.2 | UCI data sets

Apart from the artificial data sets, we also consider data sets from the UCI repository,[19] each with its own unique characteristics. Table 1 lists the data sets used, along with the initial number of target instances and outlier instances in each data set (prior to the exponential increase in the level of imbalance). All the data sets have numeric attributes and no missing values.

Apart from *alphabets* and the 3 forest data sets, all are originally binary classification problems. The alphabets data set consists of 26 classes, 1 for each letter of the English language. The forest cover data set (from which the 3 forest data sets listed above are derived) consists of 7 different classes, 1 for each unique species of trees found in the Roosevelt National Forest of Northern Colorado. To convert these into binary problems (ie, have a target and outlier class), we use aspects unique to the data set to transform them.

For the alphabets data set, the target class is represented by all instances corresponding to the letters I, J, M, and N; all other letters constitute the outlier class. The forest cover data set consists of 7

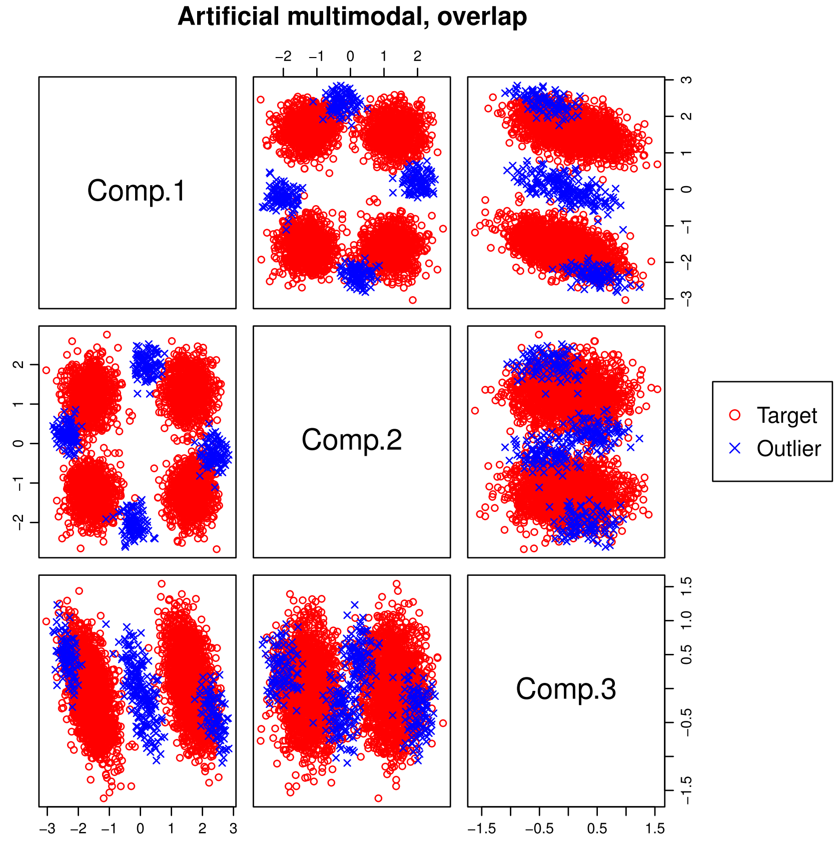


FIGURE 10 The first 3 principal components of the *multimodal artificial* data set with overlap [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 1 Description of the UCI data sets

Data set	Number of targets	Number of outliers
Diabetes	500	133
Heart disease	150	60
Ionosphere	225	63
Thyroid disease	3 541	115
Sonar	111	97
Alphabets	3 077	1538
Forest	16 500	6499
ForestC1	2 500	1249
ForestC2C5	2 400	1200

types of tree species (classes) found in the Roosevelt National Forest of Northern Colorado: spruce/fir (type 1), lodgepole pine (type 2), ponderosa pine (type 3), cottonwood/willow (type 4), aspen (type 5), Douglas fir (type 6), and krummholz (type 7). Thus, we are able to create 3 unique data sets:

Forest: This is composed of types 3, 4, 6, and 7 as the target class. While types 3, 4, and 6 form their own unique niche, the resulting distribution was found to be highly simple to learn. As a result, to add an element of complexity and multimodality, we combined the 3 with type 7.

TABLE 2 Strategies used for each data set

Data set	C	F	N
Artificial multimodal (NO)	✓	✓	✓
Artificial multimodal (O)	✓	✓	✓
Diabetes	×	×	✓
Heart disease	×	×	✓
Ionosphere	×	×	✓
Thyroid disease	×	×	✓
Sonar	×	×	✓
Alphabets	×	✓	✓
Forest	×	✓	✓
ForestC1	×	×	✓
ForestC2C5	×	×	✓

ForestC1: Type 1 is the second largest of the classes but is described as being atypical of the species found in the region. Thus, the second forest cover data set had only type 1 as the target class.

ForestC2C5: The final data set represents types 2 and 5, which are the largest distinct group in the data set, and the most typical of the region.

4.2 | Experimental framework

On the basis of the type of knowledge available over the domains, the applicable strategies are detailed in Table 2. C refers to using complete knowledge of aspects, F refers to the fuzzy knowledge approach of using supervised learning to learn the aspects during training, and N refers to the approach when we have no knowledge regarding the aspects of the domain. The fuzzy knowledge approach is applicable for the alphabets and forest domains as we know the aspects of the domain, and thus, during training, we know the subconcepts within the target class data. However, this knowledge is not available during testing. In the other UCI domains, we have no knowledge regarding the aspects of the domain, and thus, we only use the strategy for no knowledge.

We use the AA and the OCSVM for 1-class classification. The experiments with AA were implemented using the AMORE[*] R package and run in R.[†] One hidden layer was used for the AA in all the experiments, and the number of training iterations was set to 50. The momentum value was set to 0.99, and the learning rate to 0.01. The number of hidden units for the artificial data sets were set to 4. For all other data sets, they varied from 1 to the number of dimensions of the particular data set, and the number of units giving the best results was chosen. The R implementation of OCSVM via the e1071 package[‡] is used.

The performance measure we use is the geometric mean of the per-class accuracies.[20] It is given by $g\text{-}mean = \sqrt{acc_1 \times acc_2}$, where acc_i is the accuracy of the classifier on instances belonging to class i. Note that the metric is computed in a manner that is independent of imbalance, as each class is treated individually. Thus, it is immune to imbalance. Evaluation is done using a 5 × 2 cross validation.

[*] AMORE: A MORE flexible neural network package, http://cran.r-project.org/web/packages/AMORE/index.html

[†] The R Project for Statistical Computing, http://www.r-project.org/

[‡] http://cran.r-project.org/web/packages/e1071/index.html

4.3 | Results

Table 3 summarizes the results over all domains; we list the *g-mean* for the regular version of each 1-class classifier, along with the *g-mean* of the 1-class classifier under the best-performing applicable strategy (AA refers to the autoassociator, K-AA refers to the best-performing knowledge strategy for the AA, OCSVM refers to the 1-class SVM, and K-OCSVM refers to the best-performing knowledge strategy for the OCSVM). In all domains, we note that using knowledge improves the classification. In all cases, the domains exhibit complexity in terms of high overlap (all domains) and multimodality (all except diabetes and sonar).

It is worth considering where exactly the improvement in performance by using knowledge is coming from. We noted previously that learning without taking domain knowledge into account can lead to overgeneralization. The implication of this is that the majority of outlier data would be misclassified as belonging to the target class, and thus, the accuracy over the outlier class would be very poor. Using knowledge can mitigate the detrimental effect of overgeneralization. In the experiments discussed in this section, we observe that the improvement in performance comes entirely from an increase in the power of the 1-class classifier to correctly classify novel instances as outliers (ie, an improved outlier detection accuracy). In Table 4, we display the per-class accuracies for both the target and outlier classes, for the case when the 1-class classifier uses no knowledge (under the column NK) and for the best-performing knowledge strategy (under the column K).

All domains exhibit an increase in the outlier class accuracy; in some cases, the increase is highly significant. In some domains, the true-positive rate declines slightly; this is due to the classifier not overgeneralizing over the complex domain. A simple classification rule would be to accept everything as belonging to the target class, and given that we only have data from that class, this would ensure perfect accuracy over the available data. In other words, we overgeneralize over the available data. The more complex the domain, the higher the likelihood of such an overgeneralization happening. While this would ensure high accuracy over the target data, the performance over novel outlier instances would be dismal. This is evident from the results presented; all domains exhibit some level of complexity, and we observe that the outlier accuracies are relatively low compared with the target accuracies. Using domain knowledge to identify and learn along the aspects prevents this sort of overgeneralization from happening. While the target accuracy may go down slightly, by tightening the classifier, we are correctly able to reject a much larger number of outliers, which, especially in security-based domains, is extremely crucial! In practice, the bulk of data available for learning is

TABLE 3 Summary of results over all domains

Data set	AA	K-AA	OCSVM	K-OCSVM
Artificial multimodal (NO)	0.111	0.911	0.375	0.927
Artificial multimodal (O)	0.258	0.879	0.510	0.923
Diabetes	0.544	0.656	0.578	0.656
Heart disease	0.661	0.723	0.684	0.705
Ionosphere	0.794	0.909	0.894	0.910
Thyroid disease	0.579	0.663	0.552	0.675
Sonar	0.436	0.631	0.616	0.623
Alphabets	0.538	0.883	0.693	0.850
Forest	0.799	0.833	0.588	0.783
ForestC1	0.748	0.898	0.724	0.895
ForestC2C5	0.764	0.780	0.732	0.784

TABLE 4 The target and outlier class accuracies over the various domains

Data set	Classifier	Target accuracy		Outlier accuracy	
		NK	K	NK	K
Artificial multimodal (NO)	AA	0.946	0.961	0.029	0.849
	OCSVM	0.894	0.861	0.157	0.999
Artificial multimodal (O)	AA	0.950	0.941	0.074	0.827
	OCSVM	0.893	0.857	0.291	0.995
Diabetes	AA	0.924	0.703	0.308	0.656
	OCSVM	0.809	0.640	0.421	0.642
Heart disease	AA	0.898	0.748	0.490	0.628
	OCSVM	0.730	0.654	0.643	0.737
Ionosphere	AA	0.950	0.926	0.680	0.890
	OCSVM	0.930	0.912	0.860	0.890
Thyroid disease	AA	0.658	0.737	0.509	0.596
	OCSVM	0.697	0.593	0.444	0.768
Sonar	AA	0.690	0.590	0.270	0.642
	OCSVM	0.645	0.488	0.584	0.791
Alphabets	AA	0.947	0.925	0.307	0.843
	OCSVM	0.886	0.840	0.543	0.860
Forest	AA	0.897	0.886	0.712	0.787
	OCSVM	0.868	0.896	0.385	0.685
ForestC1	AA	0.893	0.876	0.626	0.920
	OCSVM	0.897	0.892	0.584	0.897
ForestC2C5	AA	0.890	0.844	0.649	0.720
	OCSVM	0.898	0.908	0.597	0.677

typically normal, or “benign”; data from anomalous classes of interest are rare or impossible to collect (eg nuclear explosions, stock market crashes, and hacking attacks). Thus, identifying a harmless instance as being harmful is much more acceptable than identifying a harmful instance as harmless. Thus, improving the detection of outliers (ie, harmful data) is paramount.

5 | PRACTICAL CASE STUDIES

In this section, we detail experiments conducted over the 3 primary domains considered in our work, namely, swipe data from smartphones, gamma ray spectra, and Xenon concentrations for CTBT verification. For each domain, we begin by describing the data, followed by the experimental framework and results. The domain of swipe-based biometrics is an example of one in which we have complete access to knowledge regarding the aspects of the domain. The domain of detecting anomalous gamma ray spectra is an example of a domain with fuzzy knowledge of the aspects, as while we can identify aspects during training, during testing, we cannot ascertain the aspect to which a novel spectrum belongs. Finally, the domain for compliance verification of the CTBT is an example of one in which we have no knowledge regarding the aspects.

5.1 | Biometrics for security: swipes

5.1.1 | Domain description

The first domain we consider is that of biometric authentication on mobile phones. Specifically, the task is to use a users *swipe* across the touch screen of a mobile device to provide authentication. When a user swipes across the screen, each sensor generates a time series; the touch screen time series represents the Cartesian coordinates of the swipe across the screen at different time intervals, and the accelerometer and gyroscope time series represent the motion of the phone in 3-dimensional space while the swipe is being done across the screen. For our study, we are only interested in biometrics defined by the users' hand motion, and thus, we only consider the accelerometer and gyroscope time series.

The accelerometer and gyroscope sensors report values for the x, y, and z axis. Our discretization methodology is illustrated in Figure 11. We begin by binning the time series into a fixed number of bins. Thus, each bin will represent a curve that represents a fixed temporal section of the time series. For each bin, we calculate 2 distinct features: the *slope* of the regression line for the curve in the bin and the *area* under the same curve. The slope represents the direction in which the time series travels in that particular time segment, whereas the area represents the magnitude of the motion. Together, these 2 values represent a fixed temporal section in the time series. Each swipe, therefore, is represented by a vector composed of the discretized accelerometer and gyroscope features. The data are collected using a custom application designed for an Android phone.

5.1.2 | Experimental framework

We identify 2 behaviors on which to split the swipe space: sitting and walking; these represent the 2 aspects of the domain. While it is possible to identify a lot more behaviors, as an initial step, we focused only on these 2. Identifying whether a user is stationary or in motion is simple; modern phone operating systems have application program interfaces that are able to detect motion. For example, one can use the *step detector* capability in the Android application program interface. Thus, when a user swipes, the phone can detect motion, or the lack of, and can process the swipe using the appropriate 1-class classifier. As a result, the learning strategy we use is *full knowledge*, as we can identify to which aspect the novel instances belong.

Two users were asked to generate these data sets, and the results are presented over these data. In particular, the mixed motion results (ie, over the whole domain) are obtained by combining all the data sets and passing them through our algorithm. The results by focusing on the aspects are obtained by passing the data sets for sitting and walking separately, and combining the results for the resulting ensemble.

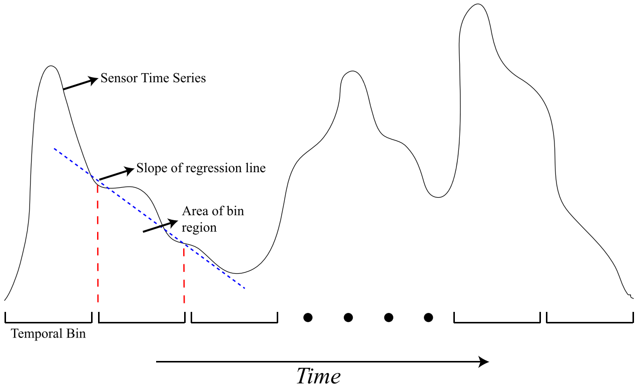


FIGURE 11 Discretization of the accelerometer and gyroscope time series [Color figure can be viewed at wileyonlinelibrary.com]

With respect to the users, user 1 swipes with 1 hand, his right, while holding the phone in the same hand. His swipes are highly consistent with respect to the touch patterns. User 2 holds the phone in both hands and swipes with the index finger as well as the thumb using his right hand. His swipes are less consistent than those of user 1. The phone used in both cases is the HTC One. We discuss 2 sets of results. In the first, we test the ability of users to authenticate themselves (the user acceptance rate). In the second set, we test the ability of our solution to defend the user against *attacks* (the attack acceptance rate); the user hands the phone over to another user, and they attempt to authenticate themselves.

5.1.3 | Results

The results from our user study experiment are presented in Tables 5 and 6. Specifically, these numbers represent the number of times the user was able to log into his phone by using their swipe without being rejected, ie, the user acceptance rate and the attacker acceptance rate, the rate at which the model authenticates attack attempts. It is worthwhile to note that typically, industry reports accuracy taken over 3 consecutive attempts; if even one of these 3 attempts is successful, the entire set is labeled as a successful attempt. We, however, only consider single attempts. Thus, our numbers, compared with industry, represent a lower bound on performance, as taking sets of 3 will increase the accuracy values. We report user acceptance rates and attacker acceptance rates separately for each motion for clarity. The results for the system over both aspects are shown under *ensemble*, whereas the results for the model built by ignoring motion are shown under *mixed*. Finally, we also present the g-mean for both systems.

If one were to observe the raw time series generated by different motions for the sensors, it will be clear that even though the user is the same, each motion corresponds to a unique aspect of the domain. This is illustrated, for sitting and walking for the user, in Figures 12 and 13 for a single-swipe *x* axis accelerometer and gyroscope, respectively.

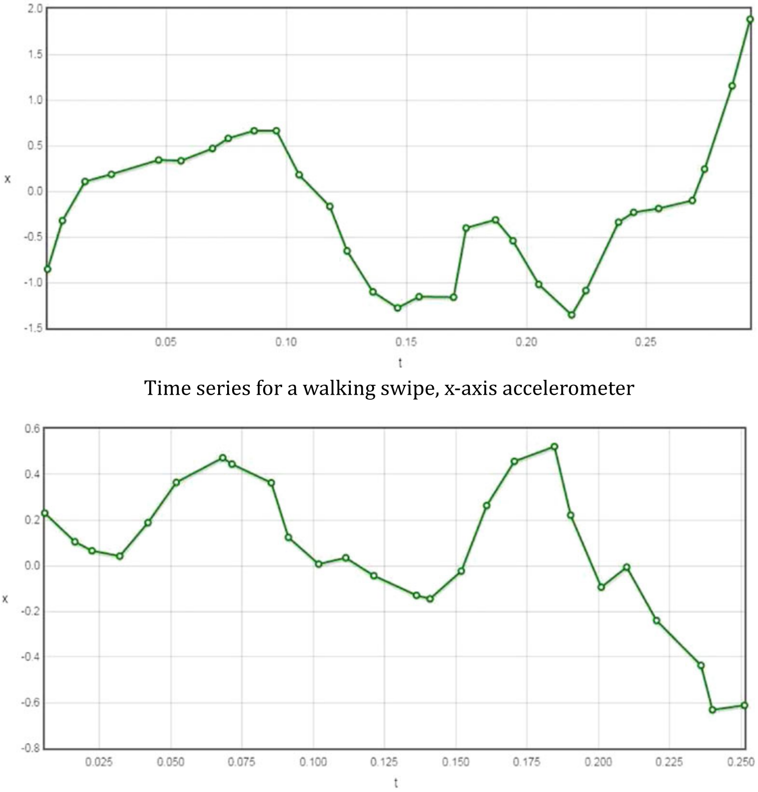
As has been the case in the domains described before, the presence of multiple subconcepts in the target data makes it very difficult to create 1-class classifiers that work well over all. This is evident by the results presented here; attempting to model both motions results in poor performance due to the model overgeneralizing over the motions. This is particularly significant for user 2, who is the more variable of the 2 users; the gains in performance are exceptionally high with respect to the attacker acceptance rate.

TABLE 5 Authentication accuracies for different motions for user 1

Motion	User acceptance rate, %	Attacker acceptance rate, %	*g*-Mean
Sitting	68.42	29.49	0.694
Walking	61.9	16.93	0.717
Ensemble	65.6	23.8	0.707
Mixed	55	39.2	0.578

TABLE 6 Authentication accuracies for different motions for user 2

Motion	User acceptance rate, %	Attacker acceptance rate, %	*g*-Mean
Sitting	71.42	12.5	0.79
Walking	54	47.36	0.533
Ensemble	63.7	27.9	0.677
Mixed	63.2	73.6	0.408



Time series for a walking swipe, x-axis accelerometer

Time series for a sitting swipe, x-axis accelerometer

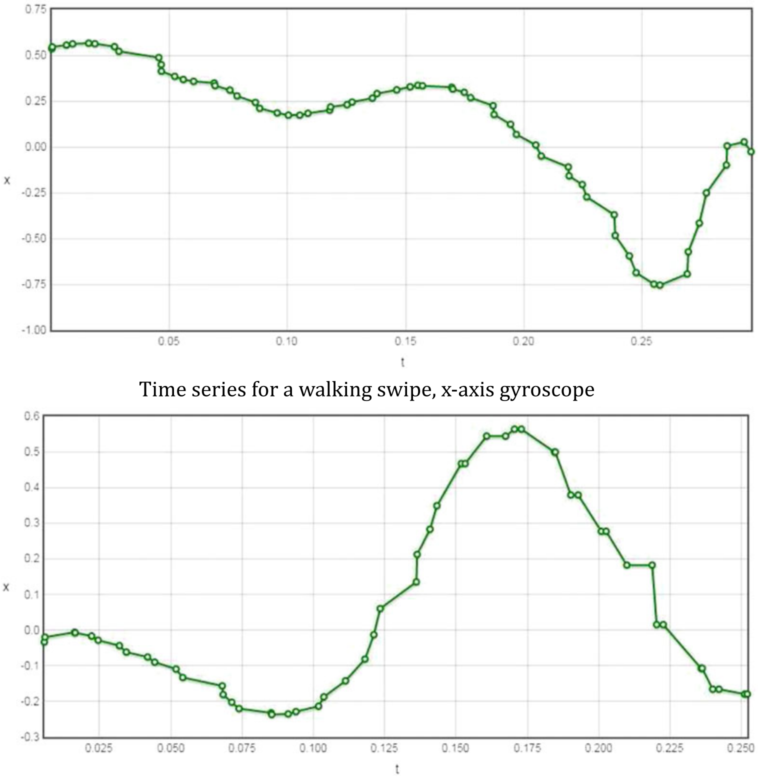
FIGURE 12 Accelerometer time series for walking and sitting [Color figure can be viewed at wileyonlinelibrary.com]

5.2 | Gamma ray spectra data

5.2.1 | Domain description

The second domain we consider pertains to invasive isotope detection over gamma ray spectra; this research was conducted in collaboration with physicists at Health Canada. Health Canada deployed 18 NaI (sodium iodide) detectors during the Vancouver Winter Olympic Games in 2010 to produce a catalog data set that can be used for future development and testing of machine learning approaches to multicategory alarm systems. The resulting measurements can be plotted as in Figures 14 and 15; the former corresponds to a pure background measurement, whereas the latter corresponds to a background plus technetium. Energy is represented in terms of channels on the x axis and the counts, which indicate the intensity, are recorded on the y axis. The isotopes of interest in our experiments all peak well below channel 600, and thus, to minimize the effects of the so-called curse of dimensionality, we only use the first 600 channels.

We were provided with data from 3 stations, and apart from the background, the readings contained spectra for 3 medical isotopes, namely, iodine, thallium, and technetium (one of the stations also had readings for caesium, which were the result of a check source). The resulting data sets displayed a prohibitive level of imbalance, as evident from the number of isotope spectra in the data from each station. These, along with the number of background instances, are enumerated below:



Time series for a walking swipe, x-axis gyroscope

Time series for a sitting swipe, x-axis gyroscope

FIGURE 13 Gyroscope time series for sitting and walking [Color figure can be viewed at wileyonlinelibrary.com]

- Station 6: 5 iodine spectra, 3 technetium spectra, and 15 caesium spectra, 39 000 background spectra
- Station 12: 2 iodine spectra, 7 technetium spectra, 25 747 background spectra
- Station 13: 3 iodine spectra, 2 technetium spectra, 2 thallium spectra, 24 709 spectra

In addition to the medical isotopes that were measured and identified as a result of people passing by the Health Canada detectors, Health Canada also provided artificially generated spectra for cobalt at varying signal strengths. These were subsequently incorporated into the data from the receptors for all stations. The sole purpose of these data was to facilitate proper evaluation; the lack of medical isotope data, while hindering the training of binary classifiers, also poses an issue for evaluation. Thus, to accurately verify the strength of the final system, these instances were used during evaluation.

5.2.2 | Experimental framework

The physicists at Health Canada indicated that the presence of rain and/or water had an impact over the gamma ray spectra measured. This can cause a greater number of dangerous isotopes to pass by undetected (high number of false positives), as it can cause the 1-class classifier to overgeneralize. Thus, the data comprise 2 distinct aspects of the spectra, based on whether they are affected by rain/water. While we have a priori knowledge of these during training, it is not possible to ascertain, during classification, which aspect the novel spectrum will belong to. This is because it is not just rain that affects a spectrum; the presence of water in the environment has an impact as well. Therefore, the

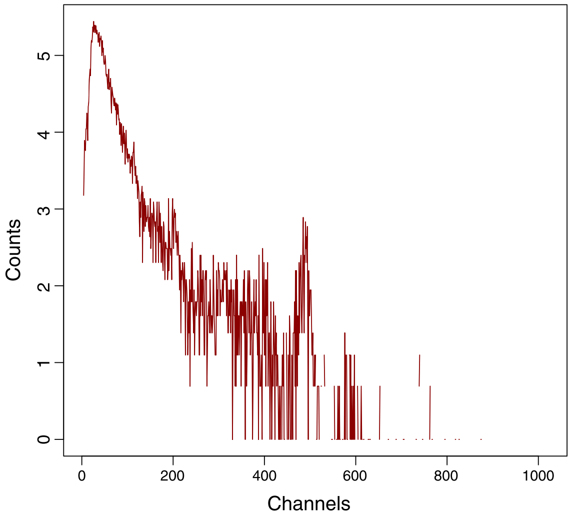


FIGURE 14 Plotted on the log scale, this depicts a background instance [Color figure can be viewed at wileyonlinelibrary.com]

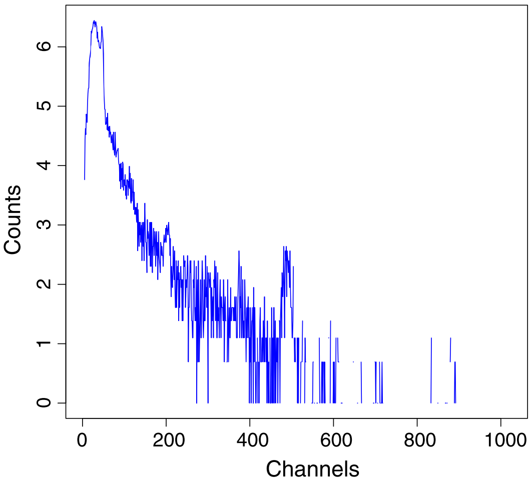


FIGURE 15 Plotted on the log scale, this depicts an instance containing the medical isotope technetium [Color figure can be viewed at wileyonlinelibrary.com]

learning strategy we use is that of *fuzzy knowledge*; we train a binary classifier to learn to discriminate between the 2 aspects and pass novel spectrum to the appropriate 1-class classifier.

With respect to labels for the training data, we were given a file in which the physicists had hand labelled the spectra based on visual inspection as well as the dose rate for the spectra. The caveat with simply using dose rate is that isotopes not impacted with the presence of water can also yield high dose rates; simply using the dose rate as the threshold is not ideal as it can result in a large number of misclassifications for isotopes. Thus, the physicists provided us with the hand-labeled file. Furthermore, the simulated cobalt data provided to us also had an ample amount of spectra impacted by the presence of water. This allowed us to train effective binary classifiers for learning the 2 aspects.

For the first tier (phase 1), we use decision trees (DTree), nearest neighbor (IBK), and naïve Bayes (NB) for discriminating between rain and nonrain aspects. They are trained on rain and nonrain spectra that consist of both background and anomalous spectra. The second tier (phase 2) consists of 2 anomaly detection systems, one for rain events and another for nonrain events. We use the Mahalanobis distance. The Mahalanobis distance relies on the calculation of a mean and a covariance matrix. These are calculated using the training data; the matrices for the rain system are calculated using the rain events from the background training set, and conversely, the matrices for the nonrain system are calculated using the nonrain events from the background training data. Note that we only use the background data for calculating the matrices, as the anomalies we expect to find are relative to the background.

The physicists desired the results as area under the receiver operating characteristic curve (AUC) values, along with the true (correctly identified background spectra) and false positives (incorrectly classified anomalous spectra). We report these results for both the system that learns over the aspects (2-tier system) and one that builds a classifier with no regard to the aspects (1-tier system). Note that we report 2 AUC values for the 2-tier system, as there are two 1-class classifiers, one for each target subconcept.

5.2.3 | Results

For the system built by learning over the aspects (2-tier system), the naïve Bayes classifier was the most effective supervised learning method for learning the aspects (rain and nonrain). For station 6, the AUC value for both rain and nonrain 1-class classifiers was 0.99. For stations 12 and 13, the AUC value over the nonrain 1-class classifier was 1, and that over the rain 1-class classifier was 0.99. For the 1-class classifier built over the entire target data with no regard to aspects (1-tier system), the AUC value was 0.99 for all 3 stations. The high AUC values for both systems can be attributed primarily to the nature of the probabilistic distribution of the individual channels of the spectral data. The near-Gaussian distribution of each channel makes the feature space of the domain exceptionally conducive to being used with the Mahalanobis distance. This in turn produces the results that we see here.

It is interesting to note that even without using a binary classifier to split testing data into rain and nonrain classes, the Mahalanobis distance–based anomaly detection system still produces exceptional AUC values. Thus, it is prudent to examine the actual true- and false-positive rates to get an idea regarding the effectiveness of the 2-tier system. This is illustrated in Table 7, where we can see that without inducing a rain separating classifier, the 1-tier system gives more than twice as many false positives as the 2-tier system. We also note that the improvement in performance is resulting, once again, in a reduction in the number of false positives due to there being less overgeneralization.

5.3 | CTBT domain

5.3.1 | Domain description

The final domain we consider pertains to data relating to the compliance verification of the CTBT. The CTBT domain was originally introduced to the machine learning community in the form of an

TABLE 7 Total number of true positives (TPs) and false positives (FPs) incurred with (2-tier system) and without (1-tier system) rain separation

	1-Tier system		2-Tier system	
Station	**TPs**	**FPs**	**TPs**	**FPs**
Station 6	3758	336	3950	144
Station 12	3856	98	3918	36
Station 13	3900	24	3914	10

open data mining competition at the International Conference on Data Mining 2008.[3] The competition invited teams to take part in building classification models from a training set that was provided by the Radiation Protection Bureau of Health Canada. For the competition, Health Canada provided data from 5 geographically distinct locations; the data set was composed of measured concentrations of ^{131m}Xe, ^{133}Xe, ^{133m}Xe, and ^{135}Xe. However, because there were no available explosion data, they provided synthesized explosion data. Stocki et al[4] used several binary classifiers to discriminate between the background and synthesized explosion data and demonstrated that these methods outperformed simple linear discriminators. Following this study, Bellinger et al[21-23] noted the "unnatural" a priori class probabilities that were inherent in the publicly available Health Canada CTBT data set, highlighting that the domain clearly fits into a 1-class classification problem. This motivated them to develop a stochastically episodic event modeling and simulation framework; the data used for our work were based on this framework.

5.3.2 | Experimental framework

Given that we have *no knowledge* regarding the aspects present in the domain, the strategy used is similar to the one used for the UCI domains. The 1-class classifiers used are the AA and the probability density estimator. The probability density estimator has also been implemented in WEKA,[24] and we use the Gaussian estimator as the density estimator and AdaBoost with decision stumps as the class probability estimator. The clustering algorithm used was the *k*-means algorithm.

The experiments with the AA were implemented using the AMORE R package and run in R. One hidden layer was used for the AA in all the experiments, and the number of training iterations was set to 50. The momentum value was set to 0.99, the learning rate was set to 0.01, and the number of hidden units ranged from 1 to the number of dimensions of the particular data set. The number of clusters varied from 2 to 20. Once again, we use the geometric mean of the per-class accuracies as the evaluation metric. It is given by $g\text{-}mean = \sqrt{acc_1 \times acc_2}$, where acc_i is the accuracy of the classifier on instances belonging to class i. Evaluation is done using stratified 10-fold cross validation.

5.3.3 | Results

The results of the 1-class classifiers over the CTBT data set are presented in Table 8. The data set has 5000 normal instances and 20 instances representing explosions.

Clustering the domain yields significant improvements in performance. This is particularly evident with the probability density estimator classifier. Clustering allows us to divide the target class into smaller subspaces (with the hope of capturing the subconcepts in the target data). Thus, the 1-class classifier induced in the cluster that contains the bulk of the outlier data will contribute to an improved overall classification, as the classifier learned over it will be far more specialized (*less generalized*) than a single classifier trained over the entire target class.

TABLE 8 Results (g-mean) of the clustered and nonclustered (normal) autoassociator and probability density estimator over the Comprehensive Test Ban Treaty data set

	g-Mean	
	Nonclustered	**Clustered**
Autoassociator	0.652	0.822
Probability density estimator	0.213	0.819

6 | CONCLUDING REMARKS

This article introduces strategies for 1-class classification that learn within the context of domain aspects, which gives rise to the presence of subconcepts within the target data. Data typically exhibit extreme levels of imbalance, making the application of binary classifiers prohibitive. Furthermore, data that are available tend to be derived from highly complex distributions; overlap between classes and multimodality in the domain can be prohibitive to the performance of 1-class classifiers. We hypothesize that these complexities arise because of the presence of subconcepts within the target data, and the performance of 1-class classifiers can be vastly improved by using domain-specific knowledge to identify and learn over these subconcepts. In particular, depending on the level of domain knowledge available, we propose 3 approaches:

1. The case when there is domain knowledge: Knowledge of the intricacies of the domain implies that we have a notion on the nature of the underlying aspects. In this case, we propose to explicitly divide the target data based on the corresponding subconcepts. Two considerations need to be taken into account in this case:
   - Identification of aspects is directly possible. In other words, it is is possible to identify the aspect to which a novel instance belongs with guaranteed 100% accuracy. This is evident in the biometric domain considered, as a mobile device's hardware is capable of identifying a user's state of motion.
   - Identification of aspects is not directly possible. However, through a domain expert, a data set having data labelled according to the domain aspects is available to enable learning them in a supervised manner. In this case, we propose inducing a multiclass classifier over the aspects; novel samples are passed to the appropriate 1-class classifier as classified by the multiclass classifier. Instead of classifying using all the 1-class classifiers in the ensemble, only the classifier that induced the target subconcept corresponding to the aspect selected by the multiclass classifier is used.
2. The case of no domain knowledge: In this case, we propose to learn the aspects in an unsupervised manner, ie, via clustering, and build 1-class classifiers over these clusters. The resulting ensemble makes the classification decision collectively.

Our work was motivated by, and tested on, real-world problems. The domains we used exemplified the variety and quality of data that are encountered in practice. The results presented in this article offer strong support for our strategies. In particular, the most significant insight to be gained is that, if possible, it is useful to exploit the nuances of the domain for producing efficient classification systems.

An important point needs to be made regarding generalizing over subconcepts. While it is not a logical absolute that all possible 1-class classifiers will fail by overgeneralizing over multiple subconcepts in the majority class, the results of our experiments over artificial and UCI domains highlight the fact that overgeneralization can be detrimental to performance when considering at least 2 powerful 1-class classifiers, the AA and the OCSVM. Furthermore, all classifiers considered in the 3 practical case studies suffer from overgeneralization as well. The simple bean illustration discussed in Section 3 illustrates subconcepts that are fairly distinct; however, in practice, subconcepts are rarely so ideally distinct. For example, in the gamma ray domain, the impact of rain causes subtle differences in the signature of the spectrum, resulting in subconcepts that are "fuzzy." Thus, while we do not claim that all classifiers *will* suffer from overgeneralization in the presence of subconcepts (indeed, such an absolute claim is impossible to verify), our work demonstrates that classifiers *can* suffer in the presence of subconcepts, and by considering the strategies in this article, we can improve classification performance.

7 | DIRECTIONS FOR FUTURE WORK

While we present 3 different strategies for 1-class classification, it is possible to combine them, depending on the application. For instance, in the complete knowledge approach, we know a priori what aspect the datum belongs too. We can use a new strategy in which we merge these 2 approaches together; we can use the power that discrimination algorithms offer to boost the complete knowledge approach. Specifically, we can induce a probabilistic classifier to assign a classification to the novel datum. Given that we have knowledge as to the aspect to which it belongs, we can check if the classification probability assigned by the classifier to the datum for that aspect is above or below a predefined threshold. If it is below the threshold, we can classify that as being an outlier, rather than further passing it along for processing by 1-class classifiers.

With respect to the strategy presented in the absence of domain knowledge, there are several interesting directions for future research into the use of clustering for 1-class classification. For the experiments presented in this thesis, we used 2 clustering algorithms: the *k*-means algorithm and the *k*-medoids algorithm. Depending on the domain, it would be worth exploring other clustering algorithms, such as the expectation-maximization algorithm, or even hierarchical clustering, for discovering clusters. The dendrogram from the hierarchical clustering could give useful insight into the presence of subconcepts in the target data and be used to determine the appropriate number of clusters to use.

The work in this article looks at binary and 1-class classifiers. One-class classifiers fit naturally as a substitute for learning problems that come under a binary formulation when data exist only for 1 class; 1 of the 2 classes becomes the target and the other the outlier. However, things are not as straightforward when the domain has multiple classes. Consider a domain with 5 classes, and data are available only from 3 classes. Given the domain formulation, it may not be prudent to consider the 2 unknown classes as belonging to a single outlier class, as knowing to which of the 5 classes a novel datum belongs could be important. Applying a 1-class classifier directly would not be appropriate, as it would treat the 2 unknown classes as a single class. Furthermore, inducing a multiclass classifier over the known classes would also not be appropriate as it would simply learn to discriminate between the 3 classes and not take into account that the domain has 2 other classes. One possibility to resolve the latter aspect would be to apply a strategy over the known classes, treating each class as a subconcept and inducing a 1-class classifier over them. However, dealing with the unknown classes would be a challenge, especially if the data are impossible to collect or reliably simulate. Thus, inducing efficient classification systems over multiclass domains that suffer from imbalance is an important and challenging avenue of research.

ACKNOWLEDGMENTS

The authors thank Rodney Berg and Kurt Ungar at Health Canada, as well as Zighra Inc. for their collaboration and financial support in our work. The work in this article was partly funded by the Canadian government through an NSERC/ISTP CRD grant.

REFERENCES

1. He H, Garcia EA. Learning from imbalanced data. *IEEE Trans Knowl Data Eng*. 2009;21(9):1263-1284.
2. Bellinger C, Sharma S, Japkowicz N. One-class versus binary classification: which and when? In: 11th Int Conf Mach Learn Appl (ICMLA), 2012, vol. 2. Boca Raton, FL: IEEE; 2012:102-106.
3. Stocki TJ, Japkowicz N, Ungar IK, Hoffman J, Yi J. Summary of the data mining contest for the IEEE International Conference on Data Mining. In: Proceedings of the ICDM'08 Data Mining Contest. Pisa, Italy: IEEE; 2008:1-6. http://www.cs.uu.nl/groups/ADA/icdm08cup/booklet.pdf

4. Stocki TJ, Li G, Japkowicz N, Ungar RK. Machine learning for radioxenon event classification for the Comprehensive Nuclear-Test-Ban Treaty. *J Environ Radioact*. 2010;101(1):68-74.
5. Japkowicz N. Class imbalances: are we focusing on the right issue? *Workshop on Learning from Imbalanced Data Sets II.* 2003:17-23.
6. Prati RC, Batista GE. APA, Monard MC. Class imbalances versus class overlapping: an analysis of a learning system behavior. *MICAI.* 2004:312-321.
7. García V, Mollineda RA, Sánchez JS. On the *k*-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Anal Appl*. 2008;11(3-4):269-280.
8. Denil M, Trappenberg T. Overlap versus imbalance. In: Adv Artif Intell. Ottawa, ON, Canada: Springer; 2010:220-231.
9. Shieh AD, Kamm DF. Ensembles of one class support vector machines. In: Multiple Classifier Syst. Reykjavik, Iceland: Springer; 2009:181-190.
10. Désir C, Bernard S, Petitjean C, Heutte L. One class random forests. *Pattern Recognit*. 2013;46(12):3490-3506.
11. Wang D, Yeung DS, Tsang ECC. Structured one-class classification. *IEEE Trans Syst Man Cybern, Part B*. 2006;36(6):1283-1295.
12. Lipka N, Stein B, Anderka M. Cluster-based one-class ensemble for classification problems in information retrieval. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval. Portland, Oregon: ACM; 2012:1041-1042.
13. Krawczyk B, Woźniak M, Cyganek B. Clustering-based ensembles for one-class classification. *Inf Sci*. 2014;264:182-195.
14. Leung K, Leckie C. Unsupervised anomaly detection in network intrusion detection using clusters. In: Proceedings of the Twenty-Eighth Australasian Conference on Computer Science, vol. 38. Australian Computer Society, Inc.: Newcastle, Australia; 2005:333-342.
15. Schwenk H, Milgram M. Transformation invariant autoassociation with application to handwritten character recognition. *NIPS*. MIT Press; 1995:991-998.
16. Giacinto G, Roli F, Didaci L. A modular multiple classifier system for the detection of intrusions in computer networks. In: Multiple Classifier Systems. Guildford, UK: Springer; 2003:346-355.
17. Giacinto G, Perdisci R, Del Rio M, Roli F. Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Inf Fusion*. 2008;9(1):69-82.
18. Sharma S, Bellinger C, Japkowicz N. Clustering based one-class classification for verification of the CTBT. In: 2012 Canadian AI. Toronto: Springer; 2012:181-193.
19. Lichman M. UCI machine learning repository. http://archive.ics.uci.edu/ml; 2013
20. Kubat M, Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the Fourteenth International Conference on Machine Learning. Nashville, TN: Morgan Kaufmann; 1997:179-186.
21. Bellinger C, Japkowicz N. Motivating the inclusion of meteorological indicators in the CTBT feature-space. In: Proceedings of IEEE Symposium on Computational Intelligence for Security and Defense Applications. Paris, France: IEEE; 2011:88-95.
22. Bellinger C, Oommen BJ. On simulating episodic events against a background of noise-like non-episodic events. In: Proceedings of the 42nd Summer Computer Simulation Conference, SCSC 2010, Ottawa, Canada; July 11–14, 2010; ACM: San Diego, CA; pp. 452-460.
23. Bellinger C, Oommen BJ. On the pattern recognition and classification of stochastically episodic events. *Transactions on Computational Collective Intelligence*. 2011;7190:1-35.
24. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH. The WEKA data mining software: an update. *ACM SIGKDD Explor Newsl*. 2009;11(1):10-18.

How to cite this article: Sharma S, Somayaji A, Japkowicz N. Learning over subconcepts: Strategies for 1-class classification. *Computational Intelligence*. 2018;34:440–467. https://doi.org/10.1111/coin.12128